UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/841,943 | 04/24/2001 | Tung Nguyen | 06356.P001 | 6228 |

7590 10/18/2006

James C. Scheller, Jr.
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Seventh Floor
12400 Wilshire Boulevard
Los Angeles, CA 90025-1026

| EXAMINER | |
|---|---|
| RYMAN, DANIEL J | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2616 | |

DATE MAILED: 10/18/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/841,943 | NGUYEN ET AL. | |
| | Examiner | Art Unit | |
| | Daniel J. Ryman | 2616 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *18 September 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-6,9-16,19-36 and 39-46* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-6,9-16,19-36 and 39-46* is/are rejected.

7)☒ Claim(s) *31* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *9/18/2006*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

## DETAILED ACTION

### *Response to Arguments*

1. Applicant's arguments with respect to claims 1-6, 9-16, 19-36, and 39-46 have been considered but are moot in view of the new ground(s) of rejection.

### *Claim Objections*

2. Claim 31 is objected to because of the following informalities: in line 31, regarding the processing of the fourth packet header, "executing a third socket software program with said first network processing offload processor, and through execution of said first network" should be" executing a *fourth* socket software program with said *second* network processing offload processor, and through execution of said *second* network". Appropriate correction is required.

### *Claim Rejections - 35 USC § 103*

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-6, 9, 10, 31-36, 39-41, 43, and 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Muller et al. (USPN 6,389,468), of record, in view of Kerr et al. (USPN 6,513,108), of record, in further view of Blount et al. (USPN 5,222,217), of record, in further view of Schoffelman et al. (USPN 6,119,170), of record, in further view of Trinh et al. (USPN 6,735,773).

5. Regarding claims 1 and 31, Muller discloses a method of processing data which is communicated over a computer network, said method comprising: pre-allocating portions of a

memory (queue) to said first processor and said second processor (see col. 50, line 60-col. 51, line 5 where each processor has its own queue); receiving first packet header data from a first network interface port (see col. 4, lines 13-14 and col. 7, lines 1-11 where it is implicit that the "high performance network interface" has a port), processing said first packet header data in a first processor which executes a first network protocol stack (see col. 9, lines 25-31 – "load distributor 112 may determine which processor an incoming packet is to be routed to for processing though an appropriate protocol stack"), and transmitting first application data associated with said first packet header data to a host processing system (see col. 12, lines 49-53 – "Ultimately, each protocol header is removed and data portion 202 is retrieved"); receiving second packet header data from said first network interface port (see col. 4, lines 13-14 and col. 7, lines 1-11 where it is implicit that the "high performance network interface" has a port), processing said second packet header data in a second processor which executes a second network protocol stack (see col. 9, lines 25-31 – "load distributor 112 may determine which processor an incoming packet is to be routed to for processing though an appropriate protocol stack"), and transmitting second application data associated with said second packet header data to said host processing system (see col. 12, lines 49-53 – "Ultimately, each protocol header is removed and data portion 202 is retrieved").

Muller further suggests processing of a third packet header data comprising receiving third application data from a host processing system and preparing said third packet header data and causing said third application data and said third packet header data to be transmitted over said computer network through said first network interface port (col. 7, lines 7-11 where the same method for "processing or transferring packets received from a network [see rejection

above] may also be used for packets moving in the reverse direction (i.e., from the NIC to the network)"); processing of a fourth packet header data comprising receiving fourth application data from said host processing system and preparing said fourth packet header data associated with said fourth application data and causing said fourth application data and said fourth packet header data to be transmitted over said computer network through said first network interface port (col. 7, lines 7-11 where the same method for "processing or transferring packets received from a network [see rejection above] may also be used for packets moving in the reverse direction (i.e., from the NIC to the network)").

Muller does not expressly disclose synchronizing access to said memory by said first and second processors. However, Muller does disclose having the processors access memory (col. 50, line 47-col. 51, line 17). Kerr teaches, in a multiprocessor system for processing packets (col. 4, lines 53-65), synchronizing access to a memory (scheduling access to the memory) by multiple processors in order to eliminate the need for arbitration between the processors for access to the memory (col. 10, lines 29-40). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to synchronize access to the memory by the first and second processors in order to eliminate the need for arbitration between the processors for access to the memory.

Muller in view of Kerr does not expressly disclose maintaining a communication channel between said first processor and said second processor through a message queue. However, Muller in view of Kerr does disclose communication among processors in order to aid in the processing of the packets (Muller: col. 50, line 47-col. 51, line 17, where having the "initial processing . . . be divided among multiple processors" requires communication between

processors). Blount teaches, in a multiprocessor system (col. 2, lines 20-22), that one well-known configuration for communication between processors in a multiprocessor system is a message queue ("storage shared among processing units into which messages from one processing unit to another processing unit may be placed") (col. 2, lines 20-26). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to maintain a communication channel between said first processor and said second processor through a message queue in order to permit communication between processors in a well-known fashion.

Muller in view of Kerr in further view of Blount suggests that the first and second processors communicate with at least one host processor of the host processing system (Muller: col. 9, line 66-col. 10, line 3 and col. 48, lines 47-56 where locating the processors on the NIC suggests that the processors are separate from the host system). However, Muller in view of Kerr in further view of Blount also discloses in other embodiments that the processors are host processors (Muller: col. 4, lines 7-38, esp. lines 32-38 and col. 9, line 66-col. 10, line 3). Schoffelman teaches, in a host processing system, that "[t]he ability to off-load communications protocol functions by using front-end processors (FEPs) is particularly advantageous for host systems" (col. 2, lines 6-24). It is implicit that one of the benefits of off-loading protocol functions to FEPs is that this frees up processing time in the host processing system. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to have the first and second processors (FEPs) communicate with at least one host processor of the host processing system since this permits the host processing system to off-load communications protocol functions, which, in turn, frees up processing time in the host processing system.

Muller in view of Kerr in further view of Blount in further view of Schoffelman does not expressly disclose that the processors execute a socket software program. In fact, Schoffelman discloses executing a socket software program on the host (Schoffelman: Fig. 2, ref. 10-204). In addition, Schoffelman discloses that the Sockets are APIs that interface the applications on the host system with protocol stacks (Schoffelman: col. 5, lines 24-37 and col. 5, lines 60-65). Trinh teaches, in a system for enabling a host to communicate with a network, interfacing host applications with a separate network processor using APIs executed on the network processor (col. 7, lines 34-64, see also col. 3, lines 16-17, which discloses that APIs are used for receiving and transmitting packets). Trinh teaches that this system is equivalent to executing the APIs on the host system (col. 7, lines 11-20). It is implicit that executing the APIs on the processor frees up computing resources in the host. As such, it would have been obvious to one of ordinary skill in the art at the time of the invention to have the processors execute a socket software program since this is a known way in which to interface a processor with a host system which frees up computing resources on the host.

6. Regarding claims 2 and 32, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh discloses that the first network protocol stack and said second network protocol stack are separate processing threads (Muller: col. 51, lines 18-38 where the individual processes on each processor are configured as "threads").

7. Regarding claims 3 and 33, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh discloses that the separate processing threads each comprise separate operating system software processing logic (Muller: col. 51, lines 18-38 where each thread operates independently, i.e. "a thread responsible for processing an incoming packet

may block itself when it has no packets to process, and awaken itself when it has work to do," such that each thread has separate operating system software processing logic to allow the thread to execute its instructions independently of the other processes).

8. Regarding claims 4 and 34, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh discloses that the first network protocol stack and said second network protocol stack use the same network protocols (Muller: col. 6, lines 35-55 where "the NIC is configured to receive and manipulate packets formatted in accordance with a protocol stack . . . supported by a network couple to the NIC" suggests that the NIC processes a single protocol stack with multiple processors on the NIC processing the same network protocol stack).

9. Regarding claims 5 and 35, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh discloses that the same network protocols comprise at least one of (a) an Internet Protocol (IP) and (b) a Transmission Control Protocol (TCP) (Muller: col. 4, lines 44-46 and col. 6, lines 43-55).

10. Regarding claims 6 and 36, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh discloses that the first group of network packets are associated with a first network session between a host processing system and a first digital processing system ("communication flow") and said second group of network packets are associated with a second network session between said host processing system and a second digital processing system ("communication flow") (Muller: col. 4, lines 7-31 and col. 13, lines 33-38).

11. Regarding claims 9 and 39, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh discloses that the first network interface port comprises an Ethernet interface (Muller: col. 6, lines 43-55).

12. Regarding claims 10 and 40, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh discloses that the first group of network packets are assigned to said first processor through a programmable hashing operation on said first group of network packets and wherein said second group of network packets are assigned to said second processor through said programmable hashing operation (Muller: col. 4, lines 19-31).

13. Regarding claims 41 and 43, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh discloses that the first network protocol stack and said second network protocol stack use different network protocols (Muller: col. 10, lines 53-61).

14. Regarding claim 44, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh suggests processing a third group of network packets in said first processor which executes said first network protocol stack, said third group of network packets being communicated through a second network interface port. Simply, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh discloses processing a group of network packets in a first processor which executes a first network protocol stack, where the group of network packets is communicated through a second network interface port (Muller: col. 4, lines 7-31). Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh also discloses that the network device can have multiple ports (Muller: col. 6, lines 63-67 and col. 6, lines 16-19). Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh further discloses

that a processor that does not have work to do will be idle (Muller: col. 51, lines 18-30). Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to process a third group of network packets in said first processor which executes said first network protocol stack, said third group of network packets being communicated through a second network interface port in order to ensure that each processor is efficiently used such that the processor is not idle.

15. Claims 11-16, 19, 20-30, 42, 45, and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Muller et al. (USPN 6,389,468), of record, in view of Kerr et al. (USPN 6,513,108), of record, in further view of Blount et al. (USPN 5,222,217), of record, in further view of Schoffelman et al. (USPN 6,119,170), of record, in further view of Trinh et al. (USPN 6,735,773) in further view of Sinks et al. (USPN 5,206,935), of record.

16. Regarding claim 11, incorporating the rejection of claims 1 and 31, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh discloses each limitation of claim 11, as outlined in the rejection of claims 1 and 31, except coupling a DMA engine and a control queue to said network interface port and said host interface port and using a bus to interconnect the various components in the device. However, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh further discloses a DMA engine and a control queue coupled to said network interface port and said host interface port (Muller: col. 53, lines 35-45 and col. 55, lines 7-20).

In addition, while Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh does not expressly disclose that a bus is used to couple the various components, Muller in view of Kerr in further view of Blount in further view of

Schoffelman in further view of Trinh does disclose the use of a bus in the system (Muller: col. 50, lines 46-54). Sinks teaches, in a multi-processor system, coupling the components of a multi-processing system using a bus (col. 1, lines 31-33; col. 1, line 68-col. 2, line 3; and col. 4, lines 37-48, where "coupled" includes direct and indirect coupling). Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to couple the various components using a bus since busses are a well known coupling device in multi-processor systems.

17. Regarding claims 12-16, 19, 20, 42, and 45, incorporating the rejection of claims 2-6, 9, 10, 32-36, 39, 40, 41, 43, and 44, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks discloses each of the limitations of claims 12-16, 19, 20, 42, and 45, as seen in the rejections of claims 2-6, 9, 10, 32-36, 39, 40, 41, 43, and 44.

18. Regarding claim 21, incorporating the rejection of claim 11, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks discloses each limitation of claim 21, as outlined in the rejection of claim 11, and in addition, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks discloses a first bus coupled to the first processor and to the second processor and to the network interface port (see e.g. Muller: col. 4, lines 7-31 and col. 7, lines 1-11 and Sinks: col. 1, lines 31-33; col. 1, line 68-col. 2, line 3; and col. 4, lines 37-48); a first memory coupled to said first bus (Muller: ref. 116: queue); a first memory controller (Muller: ref. 104: IPP) coupled to the first memory, at least a portion of said first group of network packets and a portion of said second group of network packets being stored in said first memory (Muller: col. 8, lines 23-32).

19. Regarding claim 22, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks suggests a host bus interface coupled to said first bus; a second bus coupled to said host bus interface; a second memory coupled to said second bus; a second memory controller coupled to said second bus and to said second memory; a host processor coupled to said second bus and to said second memory (Muller: col. 4, lines 7-31 and col. 7, lines 1-11 and Sinks: col. 1, lines 31-33; col. 1, line 68-col. 2, line 3; and col. 4, lines 37-48) in order to increase the number of ports connected to the host bus.

20. Regarding claim 23, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks discloses that the first processor, said second processor, said first bus and said first memory controller are all fabricated on a single integrated circuit (Sinks: col. 5, lines 5-9).

21. Regarding claim 24, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks discloses that before said first processor executes said first network protocol stack to process said first group of network packets, said portion of said first group is stored in said first memory (packet queue) (Muller: col. 55, lines 7-39). Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks discloses also discloses the use of a first direct memory access (DMA) operation to transfer data to a memory (see e.g. Muller: col. 55, lines 7-20 and Sinks: col. 2, lines 34-39). Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to use a DMA to transfer the data to the first memory since DMA is used to transfer data to a memory.

22. Regarding claim 25, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks discloses that after said first processor executes said first network protocol stack to process said first group, said portion of said first group is stored in said second memory (buffers in host memory) through a second DMA operation (Muller: col. 55, lines 7-39).

23. Regarding claim 26, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks discloses that the portion of said first group and said portion of said second group are stored in said first memory in pre-allocated portions of said first memory (Muller: col. 8, lines 23-32 and col. 55, lines 7-20).

24. Regarding claim 27, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks discloses first dispatch logic coupled to said network interface port and to said first bus, said first dispatch logic assigning said first group to said first processor through a programmable hashing operation on said first group (Muller: col. 4, lines 19-31).

25. Regarding claim 28, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks discloses that the first dispatch logic assigns said second group to said second processor through a programmable hashing operation (Muller: col. 4, lines 19-31).

26. Regarding claim 29, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks discloses second dispatch logic coupled to said first bus and to said host bus interface, said second dispatch logic assigning

packets from said second bus to one of said first processor or said second processor (Muller: col. 4, lines 19-31).

27. Regarding claim 30, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks discloses that the first processor and said second processor are general purpose, programmable processors (Muller: col. 51, lines 19-51).

28. Regarding claim 46, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks suggests that said first processor and said second processor are coupled to a further host processing system. Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks discloses that the processor is used to transfer packets from the network interface to the host (Muller: col. 7, lines 1-11). Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks also discloses that the host system can become "overburdened" with packets (Muller: col. 3, lines 12-14 and col. 3, lines 44-50). Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks further discloses that the distributing processing can yield processing gains (Muller: col. 3, lines 30-43). Thus, Muller in view of Kerr in further view of Blount in further view of Schoffelman in further view of Trinh in further view of Sinks suggests coupling the first processor and the second processor to a further host processing system in order to ensure that the host processing system can handle the high traffic loads of a fast interface.

## *Conclusion*

29. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Daniel J. Ryman whose telephone number is (571)272-3152. The examiner can normally be reached on Mon.-Fri. 8:00am-4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Huy Vu can be reached on (571)272-3155. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Daniel J. Ryman
Examiner
Art Unit 2616

DJR

HUY D. VU
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2600